**Q2**

**CODE**

import java.util.ArrayList;

import java.util.List;

class Subject {

private List<Observer> observers = new ArrayList<>();

private int state;

public int getState() {

return state;

}

public void setState(int state) {

this.state = state;

notifyAllObservers();

}

public void attach(Observer observer) {

observers.add(observer);

}

public void detach(Observer observer) {

observers.remove(observer);

}

private void notifyAllObservers() {

for (Observer observer : observers) {

observer.update();

}

}

}

abstract class Observer {

protected Subject subject;

public abstract void update();

}

class BinaryObserver extends Observer {

public BinaryObserver(Subject subject) {

this.subject = subject;

this.subject.attach(this);

}

@Override

public void update() {

System.out.println("Binary String: " + Integer.toBinaryString(subject.getState()));

}

}

class OctalObserver extends Observer {

public OctalObserver(Subject subject) {

this.subject = subject;

this.subject.attach(this);

}

@Override

public void update() {

System.out.println("Octal String: " + Integer.toOctalString(subject.getState()));

}

}

class HexaObserver extends Observer {

public HexaObserver(Subject subject) {

this.subject = subject;

this.subject.attach(this);

}

@Override

public void update() {

System.out.println("Hex String: " + Integer.toHexString(subject.getState()).toUpperCase());

}

}

public class ObserverPattern {

public static void main(String[] args) {

Subject subject = new Subject();

HexaObserver hexaObserver = new HexaObserver(subject);

OctalObserver octalObserver = new OctalObserver(subject);

BinaryObserver binaryObserver = new BinaryObserver(subject);

System.out.println("First state change: 15");

subject.setState(15);

System.out.println("Detaching OctalObserver");

subject.detach(octalObserver);

System.out.println("Second state change: 10");

subject.setState(10);

}

}